

Towards a Common Body of Knowledge on Engineering Secure Software and Services

Experiences from an EU project

Widura Schwittek

paluno – The Institute for Software Technology

University of Duisburg-Essen

SMWCon, Berlin, 22/09/2011

Agenda

- Background: NESSoS
- Requirements for a Common Body of Knowledge
- Concepts for a Common Body of Knowledge
- Realization of the Common Body of Knowledge
- Experience
- Outlook

Background



NESSoS



(NoE on Engineering Secure Software and Services)



- 12 partners from academia and industry
- Total budget: 4,7 mio. € (3,8 mio. € funded by EU)
- Duration: 42 month
- Started in 01/10/2010



- Build a long-lasting research community
 - On Engineering Secure Software and Services
 - Joining existing research communities
 - Security Engineering
 - Software Engineering
 - Service Engineering
 - Formal Methods

Background

■ Why a Common Body of Knowledge (CBK)?

→ Some **challenges** of the NoE are:

- Different communities
 - Different „bodies of knowledge“ (books, papers, mind-sets etc.)
 - Different terminologies
-
- CBK supports building and integrating a joint community

Background

- Software Engineering Body of Knowledge (SWEBOK)
→ www.swebok.org
- It is an overview of the
 - state-of-the-art
 - state-of-practice
- Create a self-understanding of the discipline
- Basis for curricula creation

Background

CBK **goals** should go one step further

- Should be created in a collaborative fashion
- Should provide additional knowledge access options
 - More than a TOC, Index, Outline along knowledge areas
 - Show all „security requirements engineering“ „methods“ that use „UML“

- Background: NESSoS

- Requirements for a Common Body of Knowledge

- Concepts for a Common Body of Knowledge

- Realization with SMW+

- Experience

- Outlook

CBK requirements

	DSpace	Liferay	ELGG	SharePoint	MyCoRe	Semantic Wiki
Creation of knowledge object collections for a specific topic and target group						wiki pages
o wiki overview page						
o learning module						
o other concepts (e.g. groups in social networks etc.)						
feedback possibilities for surrogates (meta data of uploaded documents)						
feedback possibilities for wiki pages and forum entries						
feedback possibilities for any kind of knowledge objects		for pages				
rating						
comments						
report abuse						
editing workflows: free configurable or at least three steps (e.g. draft -> accepted -> published)					only 2 steps	
roles/user groups: user, editor, administrator						
role permissions						
Repository layer						
Configurable or elaborated meta model						
Management of surrogates for documents like PDF Doc ODT XSL Multimedia (e.g. JPG AVI)						

...

⋮

⋮

CBK requirements (excerpt)

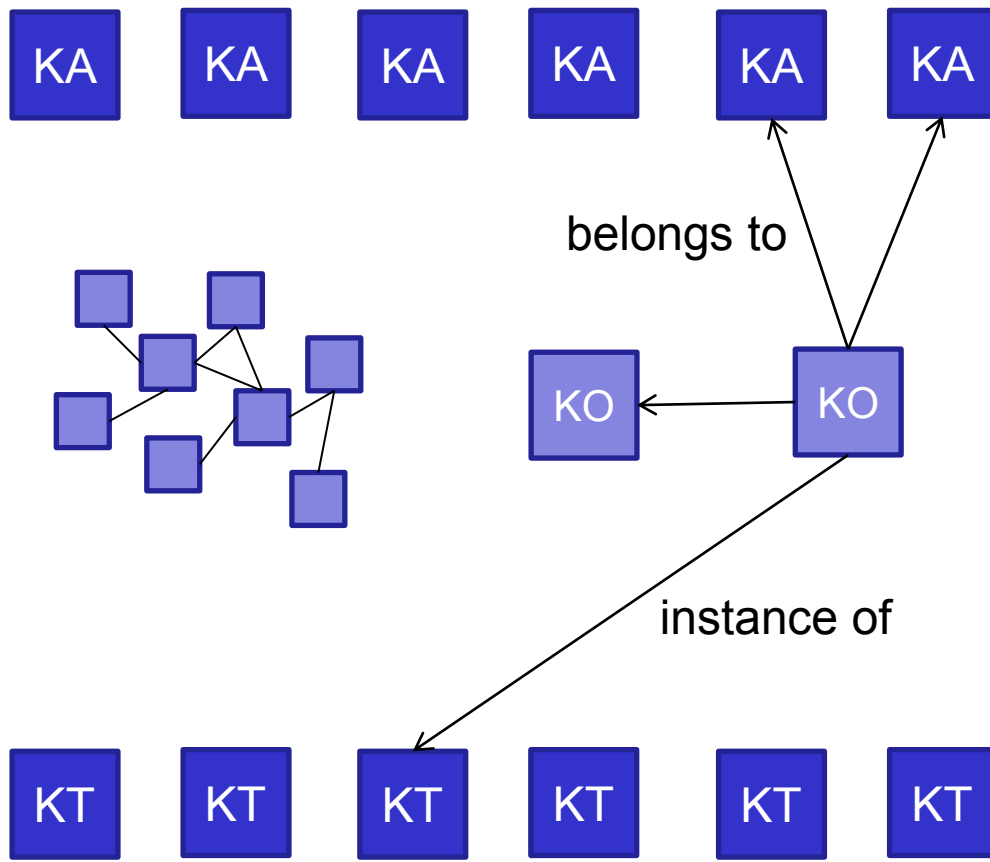
1. easy and intuitive user interface
2. collaborative content creation
3. access rights management
4. elaborate search functionality
5. mechanisms to realize SWEBOOK concepts (e.g. knowledge areas)
6. mechanisms to built up a common terminology
7. configuration rather than programming
8. adequate licensing options
9. generate a book from the CBK

→ **Semantic MediaWiki (+ Halo core ext.)**

- Background: NESSoS
- Requirements for a Common Body of Knowledge
- Concepts for a Common Body of Knowledge
- Realization with SMW+
- Experience
- Outlook

CBK concepts (1/2)

- Aligned with the SWEBOK



Knowledge Areas

(e.g. Security Requirements Engineering, Security Design)

Instances/Knowledge Objects

(e.g. secureUML, MagicUWE)

Knowledge Object Types

(e.g. Tools, Methods, Notations)

- Background: NESSoS
- Requirements for a Common Body of Knowledge
- Concepts for a Common Body of Knowledge
- Realization with SMW+
- Experience
- Outlook

Realization with SMW+

■ Representing individuals

- Knowledge Object = Ontology Individuals = Wiki pages
- Knowledge Object Type = Ontology Classes = Wiki categories
- **Template** for each Knowledge Object Type (e.g. tool, method)



MagicUWE

The CASE tool MagicUWE has been created to support the development of web applications. It focuses on the modelling phase and uses the UML-based Web Engineering (UWE) methodology. UWE provides among others a UML extension (a so called UML profile) based on stereotypes, tagged values and OCL constraints. The tool is built as a plugin for MagicDraw v.16.8. The aim is to augment usability providing additional support in the use of the web specific elements in the design, automatizing certain steps and providing shortcuts.

Contents

[hide]

- 1 Context
- 2 Problem (and motivation)
- 3 Solution
- 4 Consequences
- 5 Knowledge Area
- 6 Image Gallery
- 7 Technical Details
- 8 Usage Example(s)
- 9 Publications
- 10 Relations
- 11 Known uses
- 12 External Links

MagicUWE	
Also known as	-
KO Type	Tool
Address	Ludwig-Maximilians-Universität München Department of Computer Science Institute of Programming and Software Engineering Oettingenstr. 67 80538 Munich, Germany
Contact Email	uwe@pst.fh-lmu.de 
Website	http://uwe.pst.fh-lmu.de/toolMagicUWE.html 
Knowledge Area	Secure architecture and design, Software Quality, Security Requirements
Tags	UWE, UML-based Web Engineering, Security engineering, internet applications, Rich Internet Applications, RIA, UWEsecurity, UML
Respons. partner	LMU
Rev. Rating m8	2 - Complete
Maturity Level	(1) Proof-of-Concept/Prototype





The CASE tool MagicUWE has been created to support the development of web applications. It focuses on the modelling phase and uses the UML-based Web Engineering (UWE) methodology. UWE provides among others a UML extension (a so called UML profile) based on stereotypes, tagged values and OCL constraints. The tool is built as a plugin for MagicDraw v.16.8. The aim is to augment usability providing additional support in the use of the web specific elements in the design, automatizing certain steps and providing shortcuts.

Contents

[hide]

- 1 Context
- 2 Problem (and motivation)
- 3 Solution
- 4 Consequences
- 5 Knowledge Area
- 6 Image Gallery
- 7 Technical Details
- 8 Usage Example(s)
- 9 Publications
- 10 Relations
- 11 Known uses
- 12 External Links

MagicUWE	
Also known as	-
KO Type	Tool
Address	Ludwig-Maximilians-Universität München Department of Computer Science Institute of Programming and Software Engineering Oettingenstr. 67 80538 Munich, Germany
Contact Email	uwe@pst.ifi.lmu.de 
Website	http://uwe.pst.ifi.lmu.de/toolMagicUWE.html 
Knowledge Area	Secure architecture and design, Software Quality, Security Requirements
Tags	UWE, UML-based Web Engineering, Security engineering, internet applications, Rich Internet Applications, RIA, UWEsecurity, UML
Respons. partner	LMU
Rev. Rating m8	2 - Complete
Maturity Level	(1) Proof-of-Concept/Prototype

Context

MagicUWE is implemented as a MagicDraw plugin. It was created for Web engineers who want to model secure web applications using the UML-based Web Engineering (UWE) profile and the MagicDraw CASE tool.

Problem (and motivation)

Whenever UWE models are created, some tasks have to be repeated over and over, such for example how the navigation menu structure is built. Furthermore, some consistency checks and transformations are very time consuming if executed manually.

Solution

The plugin MagicUWE provides features like inserting UWE's stereotyped elements and copying stereotypes and their tags. Furthermore, MagicUWE supports RIA patterns, transformations between UWE models and a consistency check for secure connection redefinitions in substates or substate machines.

Consequences

MagicUWE facilitates the modeling of web applications. In particular, it provides specific elements for the modeling of security aspects, such as role based access. The advantage of UWE is to stick to the standard UML, which allows using UML CASE tools such as MagicDraw.

Realization with SMW+

- Creation of semantically enriched Wiki pages (part 1/3)
 - Semantic Forms
 - Tab extension

General	Knowledge Area	Common Terminology	Image Gallery	Technical Details	Examples	Related	Other
Also known as:	<input type="text"/>						
Postal address:	Ludwig-Maximilians-Universität München Department of Computer Science Institute of Programming and Software Engineering Oettingenstr. 67 80538 Munich, Germany						
Contact email:	<input type="text" value="uwe@pstifi.lmu.de"/>						
Website:	<input type="text" value="http://uwe.pstifi.lmu.de/toolMagicUWE.html"/>						
Executive Summary:	The CASE tool MagicUWE has been created to support the development of web applications. It focuses on the modelling phase and uses the UML-based Web Engineering (UWE) methodology. UWE provides among others a UML extension (a so called UML profile) based on stereotypes, tagged values and OCL constraints. The tool is built as a plugin for MagicDraw v.16.8. The aim is to augment usability providing additional support in the use of the web specific elements in the design, automatizing certain steps and providing shortcuts.						
Context:	MagicUWE is implemented as a MagicDraw plugin. It was created for Web engineers who want to model secure web applications using the UML-based Web Engineering (UWE) profile and the MagicDraw CASE tool.						

Realization with SMW+

- Creation of semantically enriched Wiki pages (part 2/3)
 - Multiple Instance Template for n-ary relations (e.g. tool x belongs to one or more knowledge areas)

General
Knowledge Area
Common Terminology
Image Gallery
Technical Details
Examples
Related
Other

Tool belongs to Knowledge Area(s)

Knowledge Area:
Secure architecture and design
Remove

Knowledge Area:
Software Quality
Remove

Knowledge Area:
Security Requirements

Realization with SMW+

- Creation of semantically enriched Wiki pages (part 3/3)
 - Qualified relations to relate individuals to each other
 - Autocompletion along the ontology

General
Knowledge Area
Common Terminology
Image Gallery
Technical Details
Examples
Related
Other

Publication(s)
Add another

Relation to other knowledge objects (e.g. tools, notations, methods, techniques)





Relates to knowledge objects (e.g. tools, notations, methods, techniques):
Remove
Add another

c|
Relation type:
SupportedBy

A Formal Approach to Component Adaptation
A guidance for model composition
An Expressive Aspect Composition Language for UML State Diagrams
An approach for model composition and verification
Aspect Oriented Modeling of Component Architectures Using AADL
Automatic Generation of Security Controller

Realization with SMW+

- Overview of all methods
- Semantic Queries
- Up-to-date and sortable

 Knowledge Object (KO)	 KO type	 Responsible partner	 Review Rating (m8)
MASTER Design Workbench	Tool	ATOS	1 - Incomplete
X-CREATE	Tool	CNR	1 - Incomplete
Jalapa	Tool	CNR	1 - Incomplete
WS-TAXI	Tool	CNR	1 - Incomplete
SSG: Smart & Secure GUI Builder	Tool	IMDEA	1 - Incomplete
UMLsec	Notation	IMDEA	1 - Incomplete
SecureUML	Notation	IMDEA	2 - Complete
SECTET	Notation	IMDEA	2 - Complete
EOS	Tool	IMDEA	1 - Incomplete
Aspect Oriented Modeling of Component Architectures Using AADL	Technique	INRIA	2 - Complete
Avantssar Orchestrator	Tool	INRIA	1 - Incomplete
Acr@r	Method	INRIA	1 - Incomplete
An Aspect-Oriented and Model-Driven Approach for Managing Dynamic Variability	Technique	INRIA	2 - Complete
An Expressive Aspect Composition Language for UML State Diagrams	Notation	INRIA	2 - Complete
Security-driven Model-based Dynamic Adaptation	Technique	INRIA	2 - Complete
Model-Based Software Design and Adaptation	Technique	INRIA	2 - Complete
Introducing variability into aspect-oriented modeling approaches	Technique	INRIA	2 - Complete
Verification of Access Control Requirements in Web Services Choreography	Technique	INRIA	2 - Complete
Trust Evolution Policies for Security in Collaborative Ad Hoc Applications	Technique	INRIA	2 - Complete
An approach for model composition and verification	Technique	INRIA	2 - Complete
An aspect-oriented methodology for designing secure applications	Technique	INRIA	2 - Complete
Avantssar-atos	Tool	INRIA	1 - Incomplete

Realization with SMW+

- Handbook (LaTeX)
 - Multiple semantic queries
 - Semantic Result Format: Template

Report for D5.1 (Latex format)

[\[edit\]](#)

`\section{Knowledge Area: Access Control}`

`\subsection{Methods}`

no contributions yet

`\subsection{Notations}`

no contributions yet

`\subsection{Techniques}`

`\subsubsection{Security-driven Model-based Dynamic Adaptation}`

This paper present a novel combination of Model-Driven Engineering (MDE), software product lines (SPL) and Aspect-Oriented Modeling (AOM) to support dynamic variability. SPL allows modeling variability and AOM allows the synthesis of an architecture from a choice of variants in the SPL. By composing aspects, it is possible to produce a wide range of configuration models, while managing the combinatorial explosion of variants. Using a MDE approach, they use the architecture to generate the adaptation logic needed to reconfigure the running system, instead of writing it by hand.

`\begin{table*}[tb] \centering \begin{ssftabular}{|p{.2\textwidth}|p{.8\textwidth}|}`

Realization with SMW+

Current state: [month 12](#)

■ Planning (until month 6)

- Evaluation of platforms, Initial version of structure etc.

■ Inception (month 6 – month 24)

- Closed user group
- Creation of a sound basis of contents within the NoE
- Evolution of CBK structure

■ Run (from month 24)

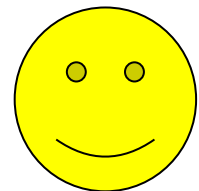
- Opened for public
- Quality assurance by moderators and user feedback

- Background: NESSoS
- Requirements for a Common Body of Knowledge
- Concepts for a Common Body of Knowledge
- Realization with SMW+
- Experience
- Outlook

Experience

- Evaluation was tedious
 - Situation changed by provided [Installer](#) and [VM](#)
- Takes time to unlock full potential
- Vibrant community
- Learn from good real-world SMW projects! (e.g. AIFB)
- Besides the technical issues mind the organizational ones!

After one year: SMW+ is still the right decision!



Goals

- Collaborative creation of a Common Body of Knowledge
- Queryable Common Body of Knowledge
- Map SWEBOK structure (Knowledge Areas, Knowledge Objects)
- Handbook generator
- **Community-driven ontology evolution**
- **Common Terminology**
- **Comparison of user-selected individuals**